Computer Modeling in Engineering & Sciences
DOI:10.32604/cmes.2020.011648
Article

Online AUV Path Replanning Using Quantum-Behaved Particle Swarm Optimization with Selective Differential Evolution

Hui Sheng Lim[1,*], Christopher K. H. Chin[1], Shuhong Chai[1] and Neil Bose[1,2]

[1]National Centre for Maritime Engineering and Hydrodynamics, Australian Maritime College, University of Tasmania, Launceston, TAS, 7250, Australia
[2]Memorial University of Newfoundland, St. John's, NL, A1C 5S7, Canada
[*]Corresponding Author: Hui Sheng Lim. Email: hui.lim@utas.edu.au
Received: 29 May 2020; Accepted: 27 July 2020

Abstract: This paper presents an online AUV (autonomous underwater vehicle) path planner that employs path replanning approach and the SDEQPSO (selective differential evolution-hybridized quantum-behaved particle swarm optimization) algorithm to optimize an AUV mission conducted in an unknown, dynamic and cluttered ocean environment. The proposed path replanner considered the effect of ocean currents in path optimization to generate a Pareto-optimal path that guides the AUV to its target within minimum time. The optimization was based on the onboard sensor data measured from the environment, which consists of a priori unknown dynamic obstacles and spatiotemporal currents. Different sensor arrangements for the forward-looking sonar and horizontal acoustic Doppler current profiler (H-ADCP) were considered in 2D and 3D simulations. Based on the simulation results, the SDEQPSO path replanner was found to be capable of generating a time-optimal path that offered up to 13% reduction in travel time compared to the situation where the vehicle simply followed a path with the shortest distance. The proposed replanning technique also showed consistently better performance over a reactive path planner in terms of solution quality, stability, and computational efficiency. Robustness of the replanner was verified under stochastic process using the Monte Carlo method. The generated path fulfilled the vehicle's safety and physical constraints, while intelligently exploiting ocean currents to improve the vehicle's efficiency.

Keywords: Autonomous underwater vehicle; path planning; particle swarm optimization; sonar detection; Monte Carlo methods

1 Introduction

AUVs have become an increasingly important tool for performing various operations, ranging from seabed surveys, coastal mapping, and environmental monitoring for scientific research purposes, to anti-submarine warfare for defense purposes. To date, most of the research has been dedicated to improving the autonomy of the AUVs in order to enable operation with longer endurance, in which the vehicles may come across unknown obstacles and large-scale time-varying ocean circulation. Strong ocean currents and eddies may push an AUV off its planned path, causing a profound impact on the vehicle's

performance, particularly its battery consumption and thus the vehicle's endurance. Therefore, it is important for an AUV path planner to take into consideration the effect of ocean currents [1]. By adapting its planning to ocean currents, a path planner can enable an AUV to surf the favorable currents that assist the vehicle's motion, while avoiding the adverse currents that are opposing it. This paper proposes a novel path replanner that generates time-optimal paths to exploit ocean currents in a fully unexplored and dynamic environment. The path replanner used an efficient SDEQPSO algorithm to achieve a balance between the generated path quality and its computational load [2], which is a crucial factor for succeeding in long-endurance missions.

Planning an AUV path in an unknown, dynamic, and cluttered underwater environment is a multi-objective and multimodal optimization problem, which requires a computationally efficient algorithm. Recent comparison studies [2–4] discussed various path planning techniques including Artificial Potential Field (APF), graphical search methods, sampling-based methods, and metaheuristic optimization. APF [5] is efficient for high dimensional problems, but it is highly vulnerable to local minima. Search-based methods such as A* [1,6] and Field D* Lite [7] are low complexity algorithms with applications limited to lower-dimensional and less complex problems. Rapidly exploring random tree (RRT) [8] and its variant RRT* [9] are sampling-based methods that can be applied for high dimensional and time-constrained scenarios, but the generated paths are usually sub-optimal and require further refinement. Metaheuristic optimization techniques such as genetic algorithm [10] and particle swarm optimization (PSO) [11] are efficient for complex multimodal path planning problems and have higher resistance towards local minima. Among the existing metaheuristic algorithms, the quantum-behaved PSO (QPSO) algorithm and its variants were found to have outstanding performance in terms of robustness and solution quality for solving the AUV path planning problem [3,12].

QPSO-based path planner is suitable for dynamic environments where real-time/online planning of the trajectory is required because it can maintain a large pool of solutions, which is available at any time during the mission. These solutions can serve as the initial solutions whenever the replanning of a path is needed, thus significantly improving the computational efficiency. Nevertheless, the algorithm may converge at local minimum solutions if the time allowed for path planning is limited, which is often the case in real AUV operations. Some have proposed methods to improve their resistance to local minima but at the cost of computational load [13]. Based on a recent comparison study [12] on the variants of the QPSO algorithm, selectively Differential Evolution (DE)-hybridized QPSO (SDEQPSO) was developed through the hybridization of DE operation in the QPSO algorithm using a selective scheme [2]. By benchmarking against other metaheuristic path planners including standard DE, PSO, and other DE-hybridized algorithms, the SDEQPSO algorithm was found to have improved search ability for the global optimal path and provide higher resistance to local minima with an insignificant increase in its computational requirement. It demonstrated the ability to generate high-quality AUV paths without imposing a high computational load on the vehicle's computer.

There are various existing techniques used to perform online path planning in an unexplored and dynamic ocean environment. The traditional approach is known as reactive path planning, which generates a new path reactively to adapt to the varying environment, while the previously planned path is discarded. To achieve online path planning while accounting for the effect of ocean currents, the reactive approach can be combined with various algorithms such as genetic algorithm (GA) [10], APF [5], level set methods [14], and swarm optimization [15]. A different approach of online path planning [16] combined path following and obstacles avoidance control to handle dynamic environment efficiently but at the cost of path quality. A similar study [17] was able to improve the path quality by combining fuzzy control and QPSO. However, this approach does not incorporate the effect of ocean currents in the path planner.

In contrast to reactive path planning, an approach known as the path replanning scheme generates a new path based on the previous solution [18]. It is deemed more computational efficient if the optimized path can

be generated by modifying the previously planned path because there is a high possibility that the new optimized path nodes can be placed near to the previous solution. This is because the ocean environmental conditions usually vary gradually over time, and therefore the new environmental conditions may resemble the conditions in the previous planning cycle to some extent. The increase in computational efficiency by making use of the previous solutions can be significant especially when the search space is vast and highly dynamic. Some existing path replanners [7,9,19] allow replanning based on a single previous solution. Path replanning can be achieved more efficiently by using a population-based optimization algorithm such as the QPSO, which can maintain all previous solutions at every iteration. For example, a QPSO-based path replanner [18] was proposed to replan the path of an AUV in a spatiotemporal environment at a predefined fixed interval. This path replanner has a high requirement for onboard sensor configuration because it requires either the global environmental information or all the information surrounding the AUV up to a certain radius to be obtained in real-time.

In this study, the novel SDEQPSO path replanner generates a time-optimal path for an AUV by adapting its solutions to ocean currents and intelligently using the currents to improve the vehicle's efficiency. Based on the onboard sensor measurements, the path replanner continuously generates the AUV path at an adaptive interval to react against the environmental changes. Different sensor configurations were considered and compared while assuming the measurements to be noise-free and reliable. The mission scenario with a priori unknown dynamic obstacles and spatiotemporal currents was first simulated in a 2D domain, followed by the simulation in a 3D domain. Monte Carlo method was used to establish a comprehensive evaluation study for the proposed path replanner. The proposed path replanner offers the following advantages:

1. It generates time-optimal paths by using a computationally efficient algorithm to improve an AUV's performance.
2. It accounts for obstacle avoidance, the spatiotemporal variability of ocean environments, and the constraints imposed by missions and vehicles.
3. No pre-generated path is required.
4. It demonstrates its scalability for missions that require different setups of onboard sensors.

This paper is organized as follows. Section 2 describes the formulation of the path planning problem. An overview of the SDEQPSO algorithm is provided in Section 3. The AUV simulation model used in this paper is outlined in Section 4. In Section 5, the simulation setup, results, and discussions are presented. Finally, the paper is concluded in Section 6 along with future research directions.

2 Problem Formulation for Path Replanning

2.1 Path Formulation

In this paper, the primary objective of the AUV path planner was to solve a multi-objective multimodal optimization problem, which was required to determine an optimal path that can guide the AUV towards its target through an ocean environment. A feasible path of the AUV can comprise a series of nodes along the path from the starting point to the endpoint (target). An optimal path can be obtained by controlling and optimizing the node coordinates. The optimization does not involve the starting point and the endpoint of the path because the same start and end locations are shared by all potential paths.

In the SDEQPSO path planner, each particle in the swarm represents a potential path solution. A swarm population containing N particles can be written as a matrix $X = [X_1, X_2,\ldots, X_N]^T$, where X denotes the position vector of the particles. Entries of the particles' position vectors represent the node coordinates. Given that every path comprises $n + 2$ nodes (including the starting point and endpoint), the path optimization involves n number of nodes. The position vector of a 2D particle requires $2n$ dimensions to register the polar coordinates of n node(s); this includes n dimension(s) for radial coordinates r and n

dimension(s) for azimuthal angle coordinates φ. Meanwhile, a 3D particle requires $3n$ dimensions to record n node(s) in the spherical coordinates, which include extra n dimension(s) for polar angle coordinates θ. The ith particle's position vectors at tth iteration for 2D and 3D can be written as Eqs. (1) and (2) respectively.

$$X_i^t = \left[x_{i,1}^t, x_{i,2}^t, \ldots, x_{i,n}^t, x_{i,n+1}^t, \ldots, x_{i,2n}^t\right], \qquad i \in \{1, 2, \ldots, N\} \tag{1}$$

$$X_i^t = \left[x_{i,1}^t, x_{i,2}^t, \ldots, x_{i,n}^t, x_{i,n+1}^t, \ldots, x_{i,3n}^t\right], \qquad i \in \{1, 2, \ldots, N\} \tag{2}$$

Cartesian coordinates of the first path node can be obtained from the polar coordinates (2D) by using Eq. (3) and from spherical coordinates (3D) by using Eq. (4).

$$\begin{aligned} x_{i,1} &= r_{i,1} \cos \varphi_{i,n+1} \\ y_{i,1} &= r_{i,1} \sin \varphi_{i,n+1} \end{aligned} \tag{3}$$

$$\begin{aligned} x_{i,1} &= r_{i,1} \cos \varphi_{i,n+1} \sin \theta_{i,2n+1} \\ y_{i,1} &= r_{i,1} \sin \varphi_{i,n+1} \sin \theta_{i,2n+1} \\ z_{i,1} &= r_{i,1} \cos \theta_{i,2n+1} \end{aligned} \tag{4}$$

B-spline geometry was applied to generate an AUV path from the path nodes. B-spline is a parametric curve generated from a series of connected piecewise polynomials [20], which are suitable for modeling the AUV path because of its continuity for a smooth path and locality for altering the path without affecting continuity. Based on the curve function in Eq. (5), B-spline curve can be constructed by using the path nodes as the control points to produce an output vector $P(u)$, which represents a $k + 1$ order B-spline curve in the form of discretized waypoints. Assuming $n + 2$ number of control points is involved, the number of piecewise polynomials that can be generated is $n + 1$.

$$P(u) = \sum_{i=0}^{n+1} x_i B_{i,k}(u), \qquad i \in \{0, 1, 2, \ldots, n+1\} \tag{5}$$

where x_i denotes the control points and u is a strictly increasing knot sequence from a knot vector $U = [u_0, \ldots, u_i, \ldots, u_{n+k+2}]$. $B_{i,k}(u)$ is the basis functions for piecewise polynomial of k degree, which can be obtained from Cox de Boor recursion [20] as follows.

$$B_{i,0}(u) = \begin{cases} 1, & \text{if } u_i \le u \le u_{i+1} \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

$$B_{i,k}(u) = \frac{u - u_i}{u_{i+k} - u_i} B_{i,k-1}(u) + \frac{u_{i+k+1} - u}{u_{i+k+1} - u_{i+1}} B_{i+1,k-1}(u) \tag{7}$$

The continuity of the spline is fully dependent on the basis functions. Hence, the control points, i.e., path nodes can be adjusted during the path optimization process without affecting the spline continuity.

2.2 Forward-Looking Sonar Model

A forward-looking sonar (FLS) model was used in the simulation for the detection of obstacles. The settings of the FLS model was specified as follows: 80 m detection range, 120° field of view, 121 number of beams (with 1° separation between beams), and 100 Hz detection frequency.

Generally, the sonar configuration of an AUV can vary depending on the mission requirements. The horizontal sonar configuration, in which the fan-shaped FLS model was installed in such a way that the

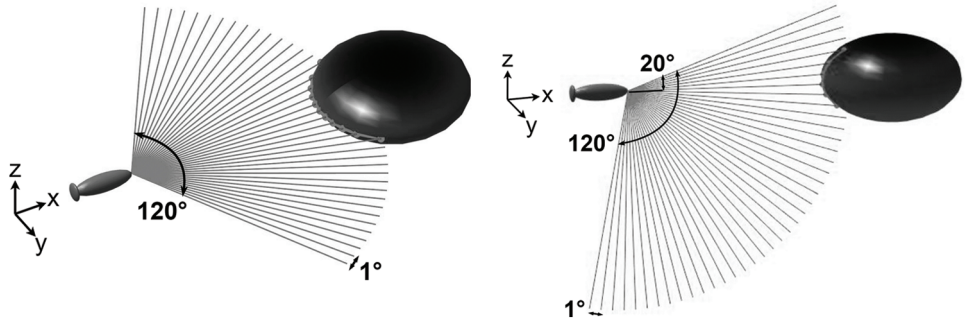

Figure 1: Forward-looking sonar configured in horizontal plane (left) and vertical plane (right)

sonar fan aligns with the horizontal plane of the vehicle, is suitable for missions such as area coverage survey. Some missions such as operations underneath ice shelves or near-seabed operations require the FLS to be configured in the vertical plane. Therefore, two sonar configurations were considered in this study as shown in Fig. 1. The vertical sonar configuration has an offset of 20° above the horizontal plane.

All obstacles in the simulated space were configured to be irregular and a priori unknown. The coordinates indicating the boundaries of the obstacles were generated by sonar detection. This information was then sent to the path planner for path computation.

2.3 Current Profiler Model

In order to allow the adaptation of path solutions to ocean currents, real-time current information based on the simulated measurement from a forward-looking horizontal acoustic Doppler current profiler (H-ADCP) was fed to the path planner. The path planning simulation used a 300 kHz H-ADCP with 200 m detection range, which is able to reconstruct a velocity profile of 200 m × 50 m in the looking-forward region of the vehicle [21].

2.4 Objective Functions

The implementation of PSO-based algorithms in an optimization problem requires the development of objective functions to evaluate the particles' fitness based on their respective solutions. Objective functions usually account for most of the computational time as PSO-based algorithms are computationally efficient [22]. Objective functions should be developed in accordance with the optimization criteria of the problem. In order to produce an accurate fitness representation model for finding the optimal solution, the developed functions must closely resemble the physical conditions of the problem space. Path planning is a minimization problem that requires the AUV travel time to be minimized. Therefore, its optimal solution should have the lowest cost/fitness. The multi-objective path planning problem was modeled using a single aggregate objective function with equal weights assigned to the underlying objective functions F_k. Thus, the optimal solution $X^{\dagger}$ for the path planning problem can be given by the function in Eq. (8).

$$X^{\dagger} = \arg\min \sum_{k=1}^{2} F_k(X_i) \tag{8}$$

The first objective function F_1 was developed to measure the particles' fitness with respect to the time required to travel on the corresponding paths while considering the effect of ocean currents. This allowed the path planner to determine the time-optimal path, which would guide the vehicle to its destination within minimum time. After constructing a B-spline path from the path nodes, the path X_i can be represented in the form of discretized waypoints $P = [p_{i,1}, p_{i,2}, \ldots, p_{i,m}]$, where P is generated from the B-spline function, and m is the total number of discretized waypoints. For the ith particle, its travel time cost

$F_1(X_i)$ can be given as the sum of discretized time required to travel on each small path segment that links the consecutive discretized waypoints in P as shown in Eq. (9).

$$F_1(X_i) = \sum_{j=1}^{m-1} \frac{\left\| \overrightarrow{p_{i,j}p_{i,j+1}} \right\|}{|V_g|}, \qquad j \in \{1, 2, \ldots, m-1\} \tag{9}$$

where V_g is the AUV's resultant ground reference velocity, which is the resultant AUV velocity under the influence of surrounding ocean currents. Projection of the current velocity V_c onto the vector of AUV water reference velocity V_a, which is in the same direction of the path vector, allowed the effect of currents on the AUV to be determined. Thus, V_g can be given by the sum of V_a and the contribution of V_c as shown in Eq. (10). Eq. (10) enabled the path planner to adapt its solutions to the measured currents.

$$V_g = V_a + \frac{V_c \cdot \overrightarrow{p_{i,j}p_{i,j+1}}}{\left\| \overrightarrow{p_{i,j}p_{i,j+1}} \right\|} \tag{10}$$

In order to generate a collision-free and feasible path, solutions generated by the algorithm were required to satisfy the following objectives and boundaries:

- *Obstacle avoidance:* Maintain a safe distance with obstacles to prevent collisions.
- *Radial boundary:* Control the placement of path nodes for path replanning.
- *Azimuthal boundary:* Constrain the solutions with respect to the AUV's minimum turning radius.
- *Polar boundary:* Constrain the solutions with respect to the AUV's pitch control limitation.

The path solutions were constrained based on a setup discussed in the previous work [23], which are explained as follows. The second objective function F_2 was designed as a penalty function for achieving obstacle avoidance. The penalty function measured the threat cost of a given path with respect to its exposure to threats/obstacles. Threat detection points, which were generated by the forward-looking sonar, were treated as circles under the 2D condition and as spheres under 3D. The radii of the threat circles/spheres were set as the safety clearance required by the AUV to maintain a safe distance with the threats. The threat cost can be obtained by checking the path's intersection with the threat circles/spheres. A threat h in 3D with a detection point $O_{c,h} = (O_{cx}, O_{cy}, O_{cz})$ and safety clearance $O_{r,h}$ can be represented by a parametric equation in Eq. (11). A path segment that connects two adjacent waypoints $p_{i,\,j} = (x_1, y_1, z_1)$ and $p_{i,\,j+1} = (x_2, y_2, z_2)$ can also be expressed as shown in Eq. (12).

$$(x - O_{cx})^2 + (y - O_{cy})^2 + (z - O_{cz})^2 = {O_r}^2 \tag{11}$$

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} + s \begin{pmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \end{pmatrix} \tag{12}$$

Substituting Eq. (12) into Eq. (11) produced the following equations, which are expressed in terms of s. The intersection between the path and the threat can be checked by computing the discriminant ξ of Eq. (13) by using Eq. (17).

$$As^2 + Bs + C = 0 \tag{13}$$

$$A = (x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2 \tag{14}$$

$$B = 2\left[(O_{cx} - x_1)(x_1 - x_2) + (O_{cy} - y_1)(y_1 - y_2) + (O_{cz} - z_1)(z_1 - z_2)\right] \tag{15}$$

$$C = (O_{cx} - x)^2 + (O_{cy} - y)^2 + (O_{cz} - z)^2 - {O_r}^2 \tag{16}$$

$$\zeta = B^2 - 4AC \tag{17}$$

There will be no intersection between the path and the threat when $\xi = 0$, i.e., the path is tangent to the threat region. When $\xi > 0$, collisions between the AUV path and the threat will occur if the roots s_1 and s_2 given by Eq. (18) are within the range of $0 \le s_1, s_2 \le 1$.

$$s_1, s_2 = \frac{-B \pm \sqrt{\xi}}{2A} \tag{18}$$

If there is a collision, the threat cost $F_2(X_i)$ can be obtained from Eq. (19), which was developed to be directly proportional to the length of the path segment contained in the threat region. The intersection points S_1 and S_2 can be determined by solving Eq. (13) using the obtained s_1 and s_2 and substituting them back into Eq. (12).

$$F_2(X_i) = \sum_{h=1}^{H} \sum_{j=1}^{m-1} \frac{\left\| \overrightarrow{S_1 S_2} \right\|}{2O_{r,h}} \tag{19}$$

In order to improve the computational efficiency during path replanning, the placement of the path nodes was constrained by the radial boundary. Each path node was constrained to be placed within a concentric annulus, which is the region bounded by a pair of adjacent concentric circles with different radii. The radii were defined by a lower boundary $R_{\min}$ and an upper boundary $R_{\max}$ as defined in Eq. (20). The search domains of radial coordinates were hard-constrained between the two boundaries.

$$\begin{aligned} R_{\min} &= \left[0, r_d, 2r_d, \ldots, r_{target}\right] \\ R_{\max} &= \left[r_d, 2r_d, 3r_d, \ldots, r_{target}\right] \end{aligned} \tag{20}$$

where r_d is defined as the radial distance between two concentric circles and r_{target} denotes the radial coordinate of the target. Path nodes exceeding the boundaries $R_{\min}$ and $R_{\max}$ will be regenerated. The total number of path nodes n required for generating the path can be controlled by r_d as defined by Eq. (21).

$$n = \left\lceil r_{target} / r_d \right\rceil. \tag{21}$$

Hard constraints were also applied to ensure the generated path solutions respect the minimum turning radius and the pitch limitation of the AUV. An azimuthal boundary φ_{max} and a polar boundary θ_{max} were used to constrain the search domain of azimuthal angle coordinate and polar angle coordinate. The path solution will fulfill the constraints if $|\varphi_{i,j}| < \varphi_{max}$ and $|\theta_{i,j}| < \theta_{max}$; otherwise, the solution will be regenerated.

2.5 Path Replanning Scheme

A path replanning scheme was employed in this paper to handle a real-time path planning scenario in a fully unknown and dynamic ocean environment. The SDEQPSO algorithm can maintain the entire population of previous solutions that can be used for replanning the path at any time throughout the mission. The path replanning process was carried out online and continuously at an adaptive interval while the AUV navigates towards its target. The adaptive replanning interval was designed to be reactive to the ocean environment, meaning that a previously optimized path will be replanned when it is unsafe or less optimal due to environmental changes. Flags that will trigger path replanning are:

- Elapsed time since the previous plan exceeds a preset threshold.
- Unexpected obstacles are detected within the safe zone of the vehicle.
- Detected obstacles intersect the previously planned path.

During path replanning, the SDEQPSO path replanner modified the previous solutions to generate a new path that is optimized for a continuously varying environment. The process of reusing the previous solution

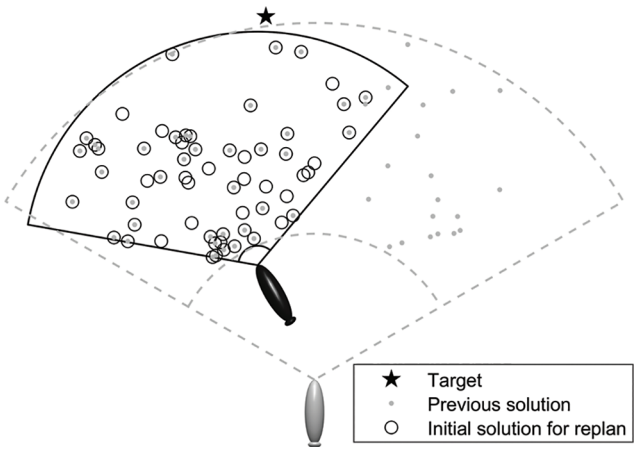

Figure 2: Reuse of solutions in path replanning process (grey vehicle denotes the previous position and black vehicle denotes the current position)

for path replanning can be described by Fig. 2, in which the grey dots and black circles represent the population of waypoints that can be used to construct the AUV path. A portion of the previous solution (grey dots) can be retained and used as the initial population (black circles) for replanning the path. The initialization of path replanning began with defining the new boundary conditions, including $R_{\min}$, $R_{\max}$, φ_{max}, and θ_{max}, based on the new starting point, which is the AUV's current position. The path waypoints behind the new starting point were removed. Next, solutions that satisfy the new boundary conditions were retained, while solutions that violate the boundary conditions were regenerated. The initialized solutions then underwent the SDEQPSO iteration to determine the optimized path.

3 SDEQPSO Algorithm

The SDEQPSO algorithm is based on the QPSO algorithm, which consists of quantum-behaved particles that search for feasible solutions by moving within a multidimensional search space. The solutions are recorded as the particles' positions. For an algorithm that contains N particles with D dimensions for solving an objective function f, the ith particle at tth iteration has the following position vector:

$$X_i^t = \left[x_{i1}^t, x_{i2}^t, \ldots, x_{ij}^t, \ldots, x_{iD}^t\right], \qquad i \in \{1, 2, \ldots, N\} \tag{22}$$

The quantum particles are assumed to be attracted to a 1-dimensional delta potential well centered at a local attraction point for each dimension of the particles' positions. In the quantum state, the momentum and energy of the particles are characterized by a wave function, and thus the position and velocity update equations of the QPSO algorithm are different from the traditional update equations in PSO. Based on the statistical interpretation of the wave function, the probability distribution function of the particles' positions can be obtained to transform the particles' positions from quantum state to classical state by employing Monte Carlo inverse transformation [22]. Accordingly, the position of the ith particle can be updated using the following stochastic equation:

$$X_i^{t+1} = \begin{cases} p_i^t + 0.5 \cdot L_i^t \cdot \ln\left(1/u_i^t\right), & \text{if } u \geq 0.5 \\ p_i^t - 0.5 \cdot L_i^t \cdot \ln\left(1/u_i^t\right), & \text{if } u < 0.5 \end{cases} \tag{23}$$

where u is a uniform distributed random positive number that is less than 1.0, p is the local attractor as defined in Eq. (24), and L is the delta potential well characteristic length as defined in Eq. (25). p and L are based on the particles' previous best position *pbest*, mean best position *mbest*, and the global best position in the swarm *gbest* as defined by Eqs. (26)–(28) respectively.

$$p_i^t = \varphi_i^t \cdot pbest_i^t + \left(1 - \varphi_i^t\right) \cdot gbest^t \tag{24}$$

$$L_i^t = 2 \cdot \beta \cdot \left| mbest^t - X_i^t \right| \tag{25}$$

$$pbest_i^t = \begin{cases} pbest_i^{t-1}, & iff(X_i^t) \geq f(pbest_i^{t-1}) \\ X_i^t, & iff(X_i^t) < f(pbest_i^{t-1}) \end{cases} \tag{26}$$

$$mbest^t = \sum_{i=1}^{N} pbest_i^t / N \tag{27}$$

$$gbest^t = \arg\min\left[f(pbest_i^t)\right] \tag{28}$$

The coefficient φ in Eq. (24) is a uniform distributed random positive number that is less than 1.0. The parameter β in Eq. (25) is known as the contraction-expansion (CE) coefficient. Combining Eqs. (23)–(25) yields the following position update equation for the particles.

$$X_i^{t+1} = \varphi_i^t \cdot pbest_i^t + \left(1 - \varphi_i^t\right) \cdot gbest^t \pm \beta \cdot \left| mbest^t - X_i^t \right| \cdot \ln\left(1/u_i^t\right) \tag{29}$$

Selection of the CE coefficient β is critical for tuning the convergence behavior of the algorithm. As suggested by an empirical study of parameter selection [22], a linearly decreasing β from a maximum value $\beta_{\max}$ of 1.0 to a minimum value $\beta_{\min}$ of 0.5 as shown in Eq. (30) is suitable for most optimization problems.

$$\beta = \beta_{\max} - \frac{t}{t_{\max}}(\beta_{\max} - \beta_{\min}) \tag{30}$$

The SDEQPSO applies DE operation on the particles through a selective scheme to increase swarm diversity and search ability without altering the original particle swarm dynamics. Following the position update operation, the particles are sorted based on their personal best position. The DE operation is applied on a selected number of particles, N_S. A selective factor S is used to control N_S according to Eq. (31).

$$N_S = N \times S, \qquad S \in [0, 1] \tag{31}$$

The DE operation includes mutation, crossover, and selection operators as described below. The mutation and crossover operators will be conducted on the N_S best-performing particles to generate the new vectors, which will replace the N_S worst-performing particles during natural selection.

- Mutation: Eq. (32) is applied to generate a mutated solution vector U.

$$U_i^t = gbest^t + \frac{(pbest_{r1}^t - pbest_{r2}^t) + (pbest_{r3}^t - pbest_{r4}^t)}{2} \tag{32}$$

where r_1, r_2, r_3 and r_4 are random particle indices that are mutually different, and different from the index i of the selected particle and the index of the global best particle, i.e., $r_1 \neq r_2 \neq r_3 \neq r_4 \neq i \neq gbest$.

- Crossover: Eq. (33) performs crossover between the mutated vector and the personal best position of the selected particle to generate a new vector T.

$$T_i^t = \left[\tau_{i1}^t, \ldots, \tau_{ij}^t, \ldots, \tau_{iD}^t\right]$$
$$\tau_{ij}^t = \begin{cases} u_{ij}^t, & \text{if } r_j \leq CR || j = r \\ pbest_{ij}^t, & \text{if } r_j > CR || j \neq r \end{cases} \tag{33}$$

where CR is the crossover probability with a suggested value of 0.85, r_j is a uniformly distributed random number in the range [0,1.0], and r is a random positive integer in the range of 1 to the total number of dimensions, D, contained by the particle.

- Natural selection: The worst-performing particle is replaced by the new vector T. All potentially optimal solutions will not be affected because only the worst-performing particles will be replaced.

For the path planning problem of an AUV, the selective factor S has a suggested value of 0.3 to help increasing swarm diversity and to maintain a sufficient number of potentially best particles [2]. The selective DE operation in SDEQPSO promotes global convergence by improving the particles' evolutionary rate and removing the least desirable solutions. The proposed SDEQPSO path replanner can be implemented according to the following pseudocode.

Step 1. **Define** the settings of algorithm and ocean environment.

Step 2. **Initialize** a group of candidate paths by generating random particle positions in Eq. (22). Define *pbest* as the current particle positions.

Step 3. **Check** the path replanning flag.

Step 4. **If** replanning flag == 1

While the termination criteria are not satisfied,

For t = 1, 2, …, t_{max},

Find *mbest* by using Eq. (27).

Obtain particle fitness $f(X_i^t)$ from the objective function.

Find *pbest* and *gbest* by using Eqs. (26) and (28) respectively.

Calculate β as required.

For each particle i = 1, 2, …, N,

Vary particle position by using Eq. (29).

End

Sort particles based on personal best fitness.

For k = 1, 2,…, N_Sth best performing particle,

Mutation: Generate mutated solution U_k^t according to Eq. (32).

Crossover: Generate trial solution T_k^t according to Eq. (33).

Natural selection: Replace kth worst-performing particle with T_k^t.

End

End

Return *gbest* that contains the optimal path upon algorithm termination.

Else

Follow the previous path.

Step 5. **Back** to Step 3 if the mission is not completed.

---

4 AUV Simulation Model

Simulation of a real-time path planning scenario requires the use of an AUV mathematical model. The SDEQPSO path planner generates the AUV path in real-time based on the feedback from the sensors and the AUV dynamic model as illustrated in Fig. 3. The generated paths were used as the reference trajectory during the simulation of a dynamic model of the Hydroid REMUS 100, 1.7-meter-long torpedo-shaped AUV. This section outlines the dynamic model and the path following controller used.

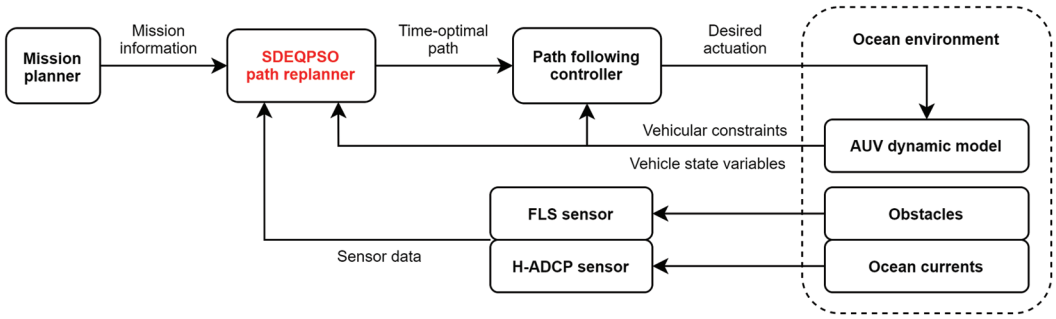

Figure 3: Implementation of SDEQPSO path replanner

4.1 Dynamic Model

The 6 DOF equations of motion of a typical AUV can be derived based on Fossen's vectorial representation [24] and SNAME (Society of Naval Architects and Marine Engineers) formulation as described in Eqs. (34) and (35).

$$\dot{\eta} = \begin{bmatrix} R(\eta_2) & 0_{3\times3} \\ 0_{3\times3} & T(\eta_2) \end{bmatrix} v \tag{34}$$

$$M\dot{v} + C(v)v + D(v)v + g(\eta) = \tau \tag{35}$$

where $R(\eta_2)$ denotes the rotation matrix of translational velocities for conversion between inertial and body-fixed reference frames and $T(\eta_2)$ is the rotation matrix of angular velocities. η includes the vehicle's position η_1 and orientation η_2 with respect to the inertial reference frame. The derivative of η in Eq. (34) represents the rate of change of η. v is the matrix that includes the vehicle's translational velocities v_1 and rotational velocities v_2 with respect to the body-fixed reference frame as shown in Eq. (36).

$$\eta = [\eta_1 \quad \eta_2]^{\mathrm{T}} = [x \quad y \quad z \quad \phi \quad \theta \quad \psi]^{\mathrm{T}},$$
$$v = [v_1 \quad v_2]^{\mathrm{T}} = [u \quad v \quad w \quad p \quad q \quad r]^{\mathrm{T}}$$

In Eq. (35), M is the inertial matrix of the rigid body and added mass, while $C(v)$ is the Coriolis matrix. $D(v)$ and $g(\eta)$ denote the hydrodynamics damping matrix and the hydrostatics restoring force respectively. The actuators' control forces are included in τ. The mathematical model of the REMUS 100 used in this study was derived from Eqs. (34) to (36) using the hydrodynamics coefficients calculated by Prestero [25].

4.2 Path Following Controller

The path following controller of the AUV model used the integral line-of-sight (iLOS) guidance law to set the yaw and pitch angles for following the generated path. The controller enables the AUV to shape its convergence towards the planned path in the presence of ocean currents and environmental disturbance by using the iLOS guidance law [26]. The desired iLOS yaw angle (heading) ψ_d and pitch angle θ_d can be determined from Eqs. (37) and (38).

$$\psi_d(e) \triangleq \arctan\left(\frac{e + K_{i,y} e_{\text{int}}}{\Delta_y}\right), \qquad K_{i,y}, \Delta_y > 0$$
$$\dot{e}_{\text{int}} = \frac{\Delta_y e}{\left(e + K_{i,y} e_{\text{int}}\right)^2 + \Delta_y^2} \tag{37}$$

$$\theta_d(h) \triangleq \arctan\left(\frac{h + K_{i,z} h_{\text{int}}}{\Delta_z}\right), \qquad K_{i,z}, \Delta_z > 0$$
$$\dot{h}_{\text{int}} = \frac{\Delta_z h}{\left(h + K_{i,z} h_{\text{int}}\right)^2 + \Delta_z^2} \tag{38}$$

where e is the cross-track error, h is the vertical-track error, $K_{i,y}$ and $K_{i,z}$ are the integral gains, and Δ_y and Δ_z represent the look-ahead distances for iLOS heading and pitch respectively. The integral terms of cross-track error e_{int} and vertical-track error h_{int} produce non-zero ψ_d and θ_d even when the AUV is on the planned path, allowing the vehicle to counteract any effects of ocean currents with the necessary sideslip and pitch angles. The rates of integral terms $\dot{e}_{int}$ and $\dot{h}_{int}$ reduce the integral action with large cross-track and vertical-track errors (i.e., vehicle is far from the planned path).

5 Simulations

The SDEQPSO path replanner is analyzed in this section. The mission objective was to determine an optimal path that guides the AUV towards a target safely and within minimum time.

5.1 Simulation Setup

The AUV mission was simulated in 2D scenarios and subsequently 3D scenarios based on the Monte Carlo method with 1000 runs. The machine used has Intel Core i5-6300U CPU @ 2.4 GHz with 8 GB RAM. The problem spaces of the simulations were assumed to be an underwater environment that contains 1000 × 1000 square grids for 2D, and 1000 × 1000 × 1000 cube grids for 3D, with a length of 1 m for each side of the grid. A priori unknown obstacles and spatiotemporal ocean currents were simulated in the problem space. The placement of the a priori unknown obstacles was configured in such a way that they will potentially block the optimized path of the AUV. In the cases of moving obstacles, they were set to move independently in different directions at random speeds up to 0.1 m/s. The variable current field with current velocity up to 0.2 m/s was generated by applying Gaussian noise to experimental data of ocean currents. The data were obtained at Beauty Point, Tasmania, Australia by using the ADCP sensors of an Explorer AUV in the University of Tasmania during one of the AUV's open water trials for the preparation of its Antarctic expedition [27].

The AUV was configured with a default water reference velocity of 1.15 m/s. Based on the properties of the REMUS 100 AUV, the safety clearance required for obstacle avoidance was defined as 3 m. The radial distance r_d was set as 50 m while the angles φ_{max} and $\theta_{\max}$ were set to 60° and 20° respectively. The test cases for the simulation are described in Tab. 1.

Table 1: Simulation test cases

Test Case	Dimension	Sonar configuration	Obstacles	Spatiotemporal currents
1	2D	Horizontal	Stationary	Yes
2	2D	Horizontal	Moving	Yes
3	3D	Horizontal	Stationary	Yes
4	3D	Horizontal	Moving	Yes
5	3D	Vertical	Stationary	Yes
6	3D	Vertical	Moving	Yes

For each run of the simulation, the maximum number of iterations for the algorithm was set to 100 with a pre-defined stopping threshold. This means the algorithm will be iterated up to a maximum number of 100 but will be stopped whenever the difference in solutions between iterations is less than the pre-set threshold. The population size of all algorithms was set to 150 particles. The setting of algorithm parameters was based on the suggested values as discussed in Section 3. The performance of the path replanner was evaluated by comparing with two other path planners:

1. SDEQPSO-based path replanner without adaptation to ocean currents,
2. SDEQPSO-based reactive path planner with adaptation to ocean currents.

Through the Monte Carlo simulation, the robustness of the planners was assessed under scenarios with stochastic processes, i.e., random-moving obstacles and random-varying ocean currents.

5.2 Simulation Results

The solutions generated by the SDEQPSO path replanner were depicted in Fig. 4. In all test cases, the mission of the AUV was to traverse the ocean field towards the target while maintaining a safe distance with the obstacles and attempting to exploit the favorable currents that would assist the AUV motion. The vehicle was driven to surf the favorable currents and to avoid the adverse currents that would oppose the vehicle's motion.

The elapsed time of the AUV mission is represented by the color bars in Fig. 4. Colors corresponding to the elapsed time are used for the planned path and the vector field of ocean currents. The boundaries of the static obstacles (Cases 1, 3 and 5) are colored brown, whereas the boundaries of the moving obstacles (Cases 2, 4 and 6) are indicated by the trails colored according to the elapsed time. Therefore, no collision will occur if the colored path does not intersect with the brown obstacles or the obstacle trails of the same color. As the obstacles were intentionally placed to block the AUV path, the AUV must detour around obstacles in all the test cases by replanning a new path whenever the previously planned paths collide with the obstacles detected by the FLS sensor. The vehicle with horizontal sonar configuration mainly maneuvered by using yaw motion, whereas the vehicle with vertical sonar configuration mostly utilized pitch motion. The resultant paths are safe and collision-free as shown in Fig. 4. During the simulation, the AUV was able to follow the planned path closely, as shown by the executed AUV paths (black lines) that closely resemble the planned paths in all test cases.

The feasibility of the path solutions can be checked by analyzing the cross-track errors of the executed paths relative to the planned paths. The calculated cross-track errors are graphed in Fig. 5. The errors for all cases were found to be well below 1 m (less than 0.1% of the total path length), proving that the AUV was able to follow the planned paths closely.

The path solutions were then validated against the vehicular constraints of the REMUS 100. The minimum turning radius of the REMUS 100 is 8.1 m in the worst-case scenario [28] and its pitch limitation is 20° (0.349 rad) based on a conservative assumption. A feasible path must have its curvature radius greater than the AUV's minimum turning radius. As shown in Fig. 6, the paths generated by the SDEQPSO satisfied the vehicle's turning and pitching constraints, validating the feasibility of the generated paths for the REMUS 100 AUV.

Next, the performances of the path replanner were assessed and compared with two other path planners based on the following properties: solution qualities, stabilities, convergence behaviors, and computational requirements. In order to study these properties, the fitness values of the obtained solutions and the computational time required to obtain the solutions were analyzed. The fitness value of a solution is given by the time required by the AUV to arrive at the target by following the generated path. Therefore, a lower fitness value is the indicator of higher solution quality and hence, a stronger search ability.

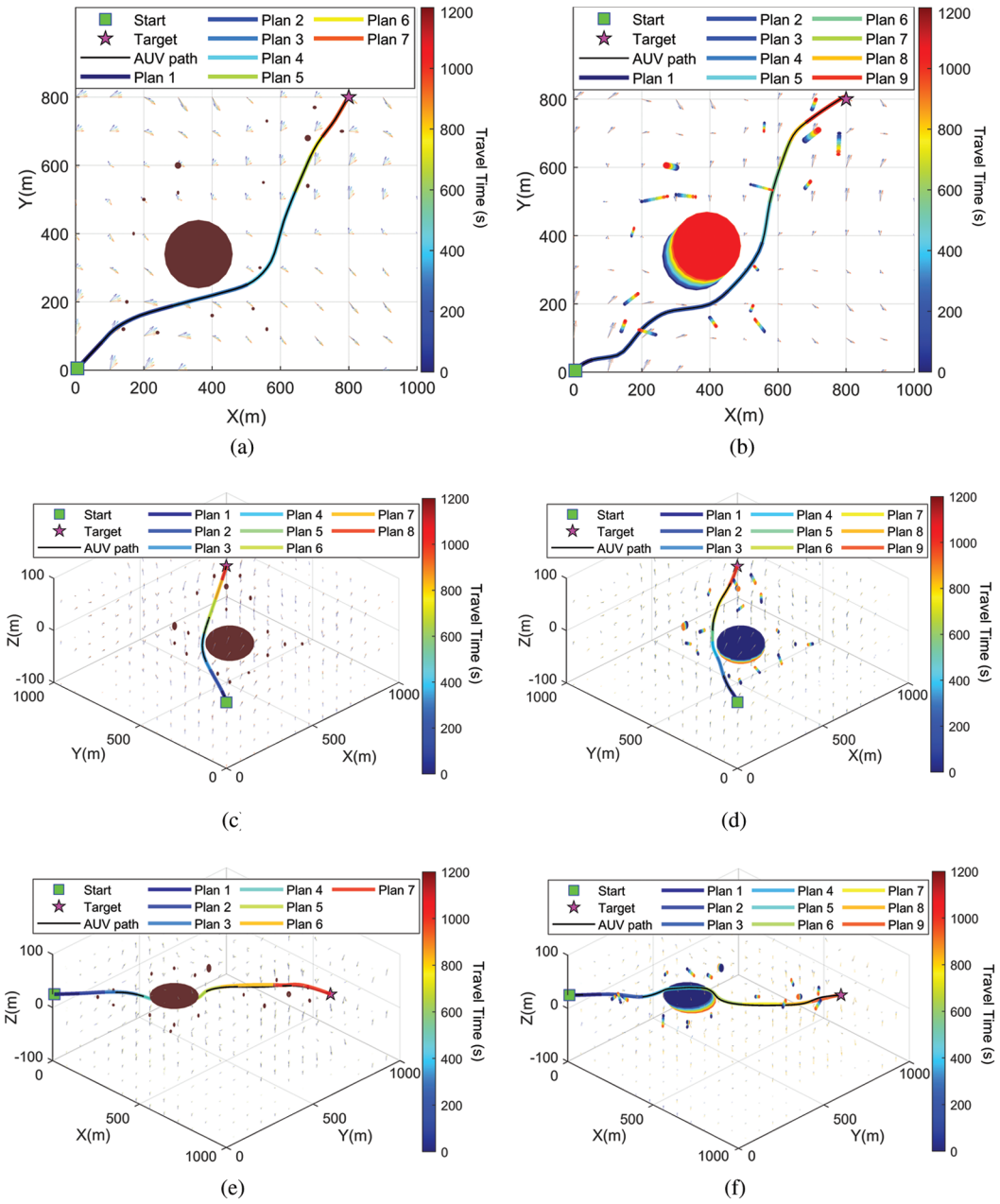

Figure 4: Pareto-optimal path solutions for (a) Case 1, (b) Case 2, (c) Case 3, (d) Case 4, (e) Case 5, and (f) Case 6

Shapiro-Wilk test with a significance level of 0.05 was used to examine the normality of the simulation results. The normality test revealed that the data was not normally distributed. Hence, medians and interquartile ranges were used as indicators for solution quality and stability. Fig. 7 shows the boxplots of the simulation results. In the boxplots, the whisker indicates the range of data. The horizontal lines inside the boxes show the medians. The upper and lower quartiles are represented by the upper and lower ends of the box, which indicates the interquartile range. The lower ends of the whiskers in the boxplot of fitness value identifies the best-known fitness for the path planners in each case.

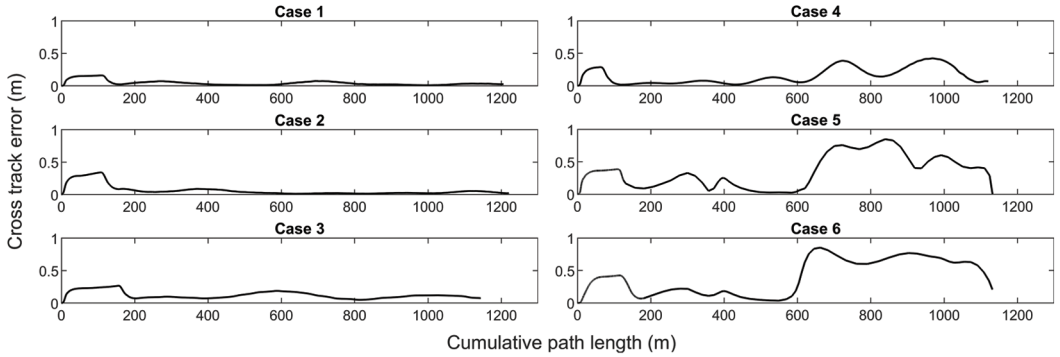

Figure 5: Variation of cross-track errors of executed paths relative to planned paths

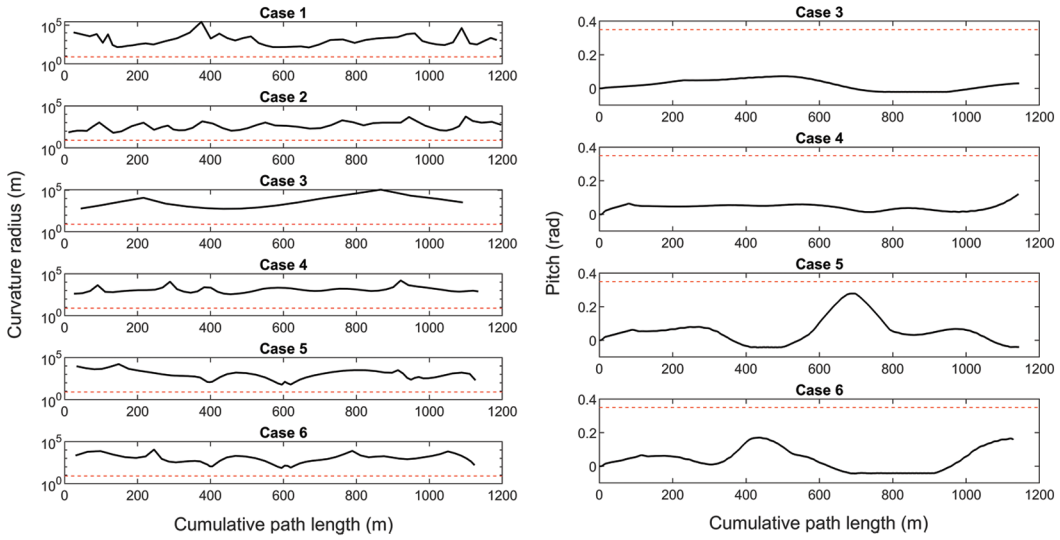

Figure 6: Variation of path curvature radius (left) and vehicle pitch (right) with respect to vehicular constraints (dashed line)

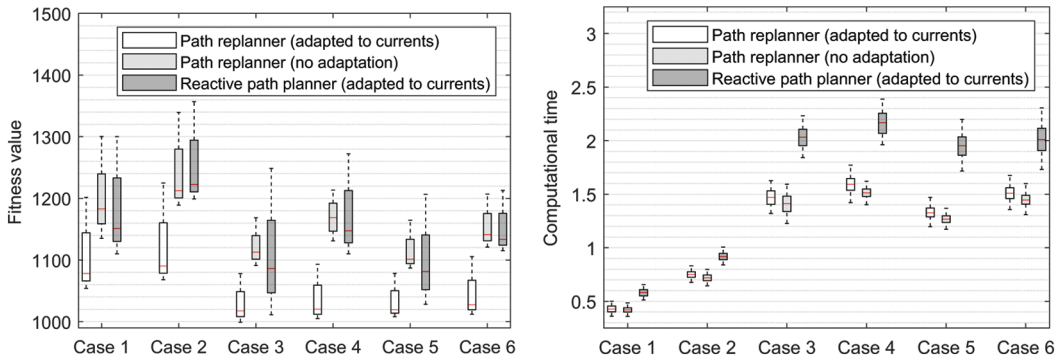

Figure 7: Fitness values (left) and computational time (right) obtained by different path planners

The effect of ocean currents on the AUV's performance was examined by comparing the proposed path replanner (adapted to currents) to the path replanner that was configured without the adaptation to currents. In contrast to the time-optimal path generated by the currents-adapted path replanner, the second replanner

simply searched for the path with the shortest distance. Fig. 7 shows that the currents-adapted path replanner generated solutions with lower medians and best-known fitness values in all test cases, suggesting a higher solution quality. The time-optimal path produced up to 13% reduction in travel time compared to the path with the shortest distance. As the shortest distance path did not take into consideration the effect of currents, the AUV that followed this path might run into adverse currents that opposed its motion and pushed it away from its path, leading to a less efficient operation. It was observed in Fig. 7 that the additional computational load for adapting the path solutions to ocean currents caused a slight increase in the computational time required by the path replanner. The simulation results showed a maximum of 5% increase in computational time, which was found to be acceptable and insignificant (less than 80 ms).

When the path replanning scheme was compared with the reactive planning, Fig. 7 shows that the medians and the best-known fitness values of the path replanner were better (lower) than the reactive planner in every test case. The path replanner was able to provide up to 11% improvement in terms of fitness values over the reactive planner, indicating the higher solution quality generated by the replanner. The path replanner was able to achieve better results because the replanning initialized the search for the optimal path from the previous solutions including the previously optimized path, whereas the reactive planner always initialized from the randomly generated solutions. This caused the reactive path planner to have poorer convergence and inadequate search before the iteration was stopped to output its final solution. In Fig. 7, it can be observed in some cases that the best-known fitness values obtained by the reactive path planner were close to the path replanner (less than 2% difference for Case 3 and Case 5). This is because the reactive path planner also used the SDEQPSO algorithm, which is a metaheuristic optimization algorithm. This means that the stochastic solutions generated by the reactive planner also have the possibility to converge at the Pareto-optimal solution although it is less likely to occur. Nonetheless, the resultant medians of fitness values produced by the reactive path planner were still worse than the replanner, leading to a significantly higher interquartile range in most test cases. The lower interquartile range of fitness values generated by the path replanner indicates higher stability and robustness in all tested scenarios.

In terms of computational time, the path replanner also outperformed the reactive path planner as shown by the shorter average time required by the replanner in all test cases. The difference in their computational time is even more significant (up to approximately 30%) when the dimension of the problem increases to 3D. The reactive path planner required longer computational time because it needs to start afresh to search for the optimal path from the randomly generated solution every time, leading to a lower rate of convergence and inefficient computation. The path replanner has faster convergence and thus shorter computational time required as a result of reusing the previous solution to effectively search for the new optimal path. The higher solution quality and shorter computational time required by the SDEQPSO path replanner indicate its higher computational efficiency. Furthermore, the consistent performance of the path replanner throughout the Monte Carlo simulation under stochastic processes verifies its robustness in generating a safe and feasible AUV path.

6 Conclusion

In this study, the SDEQPSO algorithm has successfully employed for an online path replanner of an AUV in a dynamic operational environment. Using the onboard measurements from various sensor configurations, the proposed path replanner incorporated the effect of ocean currents in path optimization to continuously generate a time-optimal path for the AUV throughout its mission. Based on the simulation results, the time-optimal path generated by the proposed path replanner offered up to 13% reduction in travel time compared to a path replanner that neglected the effect of currents. The proposed path replanning technique was also proven to have better performance over a reactive path planner in terms of solution quality (provides up to 11% improvement in fitness values), stability and computational

efficiency (provides up to 30% reduction in computational time). With the verified robustness through the Monte Carlo method, the generated path fulfilled the vehicle's safety and vehicular constraints, while taking into consideration the effect of ocean currents to improve the vehicle's operational efficiency. The future extension of this work will include incorporating noise in the sensor measurements during the simulation. Hardware-in-the-loop simulation in the AUV control software can also be applied to further evaluate the performance of the path replanner during the mission planning stage prior to actual AUV trials.

Acknowledgement: The authors acknowledge Autonomous Maritime Systems Laboratory (AMSL) in the Australian Maritime College (AMC) for providing the data from the open water trial conducted in July 2017 at Beauty Point, Tasmania, Australia.

Funding Statement: The present work is supported by a Tasmania Graduate Research Scholarship provided by AMC, University of Tasmania.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Huynh, V. T., Dunbabin, M., Smith, R. N. (2015). Predictive motion planning for AUVs subject to strong time-varying currents and forecasting uncertainties. *IEEE International Conference on Robotics and Automation,* Seattle, WA, 1144–1151.
2. Lim, H. S., Fan, S., Chin, C. K. H., Chai, S., Neil, B. (2020). Particle swarm optimization algorithms with selective differential evolution for AUV path planning. *International Journal of Robotics and Automation, 9(2),* 94–112. DOI 10.11591/ijra.v9i2.pp94-112.
3. Zeng, Z., Sammut, K., Lian, L., He, F., Lammas, A. et al. (2016). A comparison of optimization techniques for AUV path planning in environments with ocean currents. *Robotics and Autonomous Systems, 82,* 61–72. DOI 10.1016/j.robot.2016.03.011.
4. Youakim, D., Ridao, P. (2018). Motion planning survey for autonomous mobile manipulators underwater manipulator case study. *Robotics and Autonomous Systems, 107,* 20–44. DOI 10.1016/j.robot.2018.05.006.
5. Kruger, D., Stolkin, R., Blum, A., Briganti, J. (2007). Optimal AUV path planning for extended missions in complex, fast-flowing estuarine environments. *Proceedings IEEE International Conference on Robotics and Automation,* Roma, Italy, 4265–4270.
6. Koay, T. B., Chitre, M. (2013). Energy-efficient path planning for fully propelled AUVs in congested coastal waters. *MTS/IEEE OCEANS,* Bergen, Norway, 1–9.
7. Sun, B., Zhu, D. (2016). Three dimensional D∗ Lite path planning for autonomous underwater vehicle under partly unknown environment. *12th World Congress on Intelligent Control and Automation,* Guilin, China, 3248–3252.
8. Rao, D., Williams, S. B. (2009). Large-scale path planning for underwater gliders in ocean currents. *Australasian Conference on Robotics and Automation,* Sydney, Australia, 2–4.
9. Hernández, J. D., Vidal, E., Moll, M., Palomeras, N., Carreras, M. et al. (2019). Online motion planning for unexplored underwater environments using autonomous underwater vehicles. *Journal of Field Robotics, 36(2),* 370–396. DOI 10.1002/rob.21827.
10. Alvarez, A., Caiti, A., Onken, R. (2004). Evolutionary path planning for autonomous underwater vehicles in a variable ocean. *IEEE Journal of Oceanic Engineering, 29(2),* 418–429. DOI 10.1109/JOE.2004.827837.
11. MahmoudZadeh, S., Yazdani, A. M., Sammut, K., Powers, D. M. (2018). Online path planning for AUV rendezvous in dynamic cluttered undersea environment using evolutionary algorithms. *Applied Soft Computing, 70,* 929–945. DOI 10.1016/j.asoc.2017.10.025.
12. Lim, H. S., Fan, S., Chin, C. K. H., Chai, S. (2018). Performance evaluation of particle swarm intelligence based optimization techniques in a novel AUV path planner. *IEEE OES Autonomous Underwater Vehicle Symposium,* Porto, Portugal, 1–7.

13. Fu, Y., Ding, M., Zhou, C., Hu, H. (2013). Route planning for unmanned aerial vehicle (UAV) on the sea using hybrid differential evolution and quantum-behaved particle swarm optimization. *IEEE Transactions on Systems, Man, and Cybernetics: Systems, 43(6),* 1451–1465. DOI 10.1109/TSMC.2013.2248146.
14. Lolla, T., Ueckermann, M., Yiğit, K., Haley, P. J., Lermusiaux, P. F. (2012). Path planning in time dependent flow fields using level set methods. *2012 IEEE International Conference on Robotics and Automation,* Saint Paul, MN, 166–173.
15. Witt, J., Dunbabin, M. (2008). Go with the flow: optimal AUV path planning in coastal environments. *Australasian Conference on Robotics and Automation,* Canberra, Australia, 1–9.
16. Yao, X., Wang, X., Wang, F., Zhang, L. (2020). Path following based on waypoints and real-time obstacle avoidance control of an autonomous underwater vehicle. *Sensors, 20(3),* 795. DOI 10.3390/s20030795.
17. Sun, B., Zhu, D., Yang, S. X. (2018). An optimized fuzzy control algorithm for three-dimensional AUV path planning. *International Journal of Fuzzy Systems, 20(2),* 597–610. DOI 10.1007/s40815-017-0403-1.
18. Zeng, Z., Sammut, K., Lammas, A., He, F., Tang, Y. (2015). Efficient path re-planning for AUVs operating in spatiotemporal currents. *Journal of Intelligent & Robotic Systems, 79(1),* 135–153. DOI 10.1007/s10846-014-0104-z.
19. Galceran, E., Campos, R., Palomeras, N., Ribas, D., Carreras, M. et al. (2015). Coverage path planning with real-time replanning and surface reconstruction for inspection of three-dimensional underwater structures using autonomous underwater vehicles. *Journal of Field Robotics, 32(7),* 952–983. DOI 10.1002/rob.21554.
20. Piegl, L., Tiller, W. (2012). *The NURBS book*. Berlin: Springer Science & Business Media.
21. Garau, B., Alvarez, A., Oliver, G. (2006). AUV navigation through turbulent ocean environments supported by onboard H-ADCP. *Proceedings 2006 IEEE International Conference on Robotics and Automation,* Orlando, FL, 3556–3561.
22. Sun, J., Lai, C. H., Wu, X. J. (2012). *Particle swarm optimisation classical and quantum perspectives*. Boca Raton, FL: CRC Press.
23. Lim, H. S., Fan, S., Chin, C. K. H., Chai, S., Bose, N. et al. (2019). Constrained path planning of autonomous underwater vehicle using selectively-hybridized particle swarm optimization algorithms. *IFAC-PapersOnLine, 52(21),* 315–322. DOI 10.1016/j.ifacol.2019.12.326.
24. Fossen, T. I. (1999). *Guidance and control of ocean vehicles*. Norway: John Wiley & Sons.
25. Prestero, T. T. J. (2001). *Verification of a six-degree of freedom simulation model for the REMUS autonomous underwater vehicle*. Cambridge, MA: Massachusetts Institute of Technology.
26. Caharija, W., Pettersen, K. Y., Gravdahl, J. T., Børhaug, E. (2012). Path following of underactuated autonomous underwater vehicles in the presence of ocean currents. *IEEE Conference on Decision and Control,* Maui, HI, 528–535.
27. Pyper, W. (2018). Yellow submarine prepares for first Antarctic mission. *Australian Antarctic Magazine,* 12–13.
28. Eng, Y., Teo, K. M., Chitre, M., Ming Ng, K. (2016). Online system identification of an autonomous underwater vehicle via in-field experiments. *IEEE Journal of Oceanic Engineering, 41(1),* 5–17.